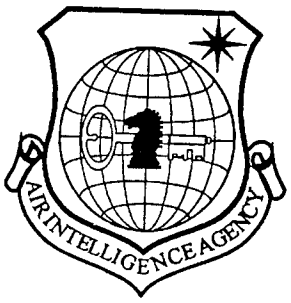
NAIC-ID(RS)T-0921-92

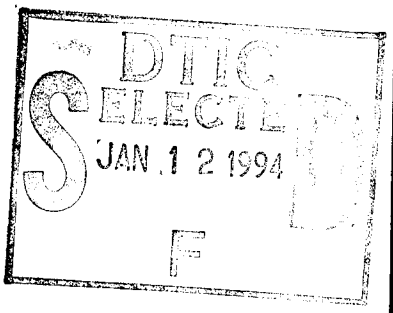
NATIONAL AIR INTELLIGENCE CENTER



THE APPLICATION OF ADAPTIVE MESH REFINEMENT TO THE NUMERICAL SIMULATION OF COMPRESSIBLE FLOW

by

Shen Jianxiong, Huang Lanjie



HUMAN TRANSLATION

NAIC-ID(RS)T-0921-92 17 November 1994

MICROFICHE NR: 94C000491

THE APPLICATION OF ADAPTIVE MESH REFINEMENT TO THE NUMERICAL SIMULATION OF COMPRESSIBLE FLOW

By: Shen Jianxiong, Huang Lanjie

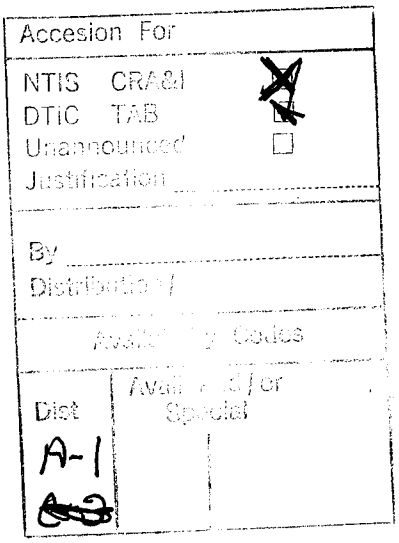
English pages: 16

Source: Kongqidonglixue Xuebao, Vol. 9, Nr. 3, September 1991; pp. 330-337

Country of origin: China
Translated by: SCITRAN
F33657-84-D-0165
Quality Control: Ruth A. Peterson
Requester: NAIC/TATV/Paul F. Freisthler

GRAPHICS DISCLAIMER

All figures, graphics, tables, equations, etc. merged into this translation were extracted from the best quality copy available.



THE APPLICATION OF ADAPTIVE MESH REFINEMENT TO THE NUMERICAL SIMULATION OF COMPRESSIBLE FLOW[1]

Shen Jianxiong Huang Lanjie

/330*

ABSTRACT

This article is based on the adaptive mesh refinement calculation methods developed by M.J.Berger and J.Oliger [2,8]. It opts for the use of the concept of numerous units formed into a grid, used in solving hyperbolic type equation sets. It combines finite difference forms, opts for the use of Richardson extrapolation techniques to automatically carry out local truncation error estimates, and, for regions with low accuracy, produces new fine mesh local refinements or eliminates old fine mesh refinements which are no longer needed, in order to reach, in the minimum amount of operations, the specified accuracy requirements. Grids are capable of going down into a layer on layer refinement. On the basis of the layered sequence of coverage, each individual grid is a rectangular uniform grid or mesh in any direction desired. This set of algorithms is independent of difference forms used in solutions, is very easy, and combines various types of forms.

We set out from the algorithm or program of M.J.Berger and J.Oliger [2,8]. We made reference to M.J.Berger and P.Colella [6] on conservation type equation sets and realized numerical value flux conservation on the interfaces of coarse and fine meshes or grids. From this, it is then possible to calculate discontinuous solutions. As far as grid boundary treatments, error estimates, and the formation of new grids are concerned, some progress has been made. Moreover, it is possible, in a flexible manner, to specify refinements in any subdomain which is of interest.

We use unsteady Euler equation sets to calculate Mach 3 flow movements in two dimensional tubes having forward facing steps. We opted for the use of MacCormack explicit difference forms with the addition of artificial viscosity. Altogether, we used a four layer grid structure: Δ = 1/10, 1/20, 1/80, and 1/320. Among these, the basic grid is Δ = 1/10. When one gets to the steps, the first instance of shock wave reflection subdomains is refined from Δ = 1/20 to Δ = 1/80. The amount of operations needed to calculate to steady states is only equal to 28% of that

---

[1] This article was received on 23 October 1989. This was a project helped with funding from the National Natural Sciences Fund and High Technology funding assistance.

* Numbers in margins indicate foreign pagination. Commas in numbers indicate decimals.

for a uniform grid of $\Delta = 1/80$. We also used $\Delta = 1/320$ grids in refinements specified for step corners and locations of tangential direction discontinuities. The results clearly show that it is possible to make clear improvements in shock wave Mach refections associated with steps. Tangential direction discontinuities are clearly distinguishable. The amount of operations required is 45% of that for the uniform $\Delta = 1/80$ grid discussed above.

INTRODUCTION

In partial differential equation calculations, using finite difference methods or finite element methods, normally, the calculation grid is predetermined, fixed, and unchanging. The accuracy of the discrete breakdown depends on the grid or mesh interval. However, normally, in this type of situation, the initial time breakdown is a smooth one, following along with the advance of the time duration. In local subdomain breakdowns, changes are very great. This even goes to the point of producing discontinuity breakdowns. This is due to limitations on computer speeds and storage. Calculation grids are not capable of infinite refinement. We are only able to hope for the use of fine mesh calculations in these local subdomains, and, as far as smooth portions are concerned, still using coarse grid calculations. In recent years, the development of zonal methods or domain decomposition methods takes the calculation subdomain as a whole and divides it into a few sections. In each individual section, it is possible to use different grids and calculation methods for independent calculations [13,16].

/331

However, in the case of phenomena which one does not know are going to occur beforehand, it is necessary, in the process of calculations, to automatically use fine mesh calculations on localized breakdowns of subdomains with great changes and to make errors for the calculation subdomain as a whole consistent with each other. In recent years, the development of adaptive mesh methods has been rapid.

Generally speaking, adaptive mesh methods are divided into two types: "universal" or "local" methods. The first type of universal methods (or moving grid point methods) refers to a fixed total number of calculation subdomain grid points. Going through certain standard movements, grid points move forward to optimum distributions in order to obtain highly accurate solutions for the entire calculation subdomain. At the same time, there is a requirement for the grid to be smooth and the grid lines, to as great an extent as possible, to be orthogonal [9,14,17,21,22]. The adaptive standards which are used in universal methods are, at the present time, of many and various types. However, generally speaking, they are not sufficiently accurate. Moreover, they also require opting for the use of arbitrary parameters and smooth functions in order to guarantee grid quality.

The other type of local methods (or local mesh refinement methods) refer to a total number of grid points in the calculation subdomain which is not fixed. As far as given accuracy requirements are concerned, one goes through a minimum increase in grid points in order to make the whole calculation subdomain breakdown satisfy accuracy requirements. As far as local mesh refinement methods are concerned, there is one type which adds refinement, point by point, to grid points with large errors and embeds fine mesh into coarse grids [1,18,20]. The data structure of this type of method is complicated. It is not suitable for use with parallel calculations. There is also a type which adds block refinements to grid points with large errors. Fine mesh refinements are overlaid on the tops of coarse grids. Except for coarse and fine grid or mesh interface points, coarse and fine grid or mesh calculations are done on an individual and independent basis. Professor J.Oliger of Stanford University and his students, W.D.Gropp, J.H.Bolstad, M.J.Berger, S.C.Caruso, and W.Skamarock, et al, developed just this type of method [2,6-8,10-12,18,19,23,24]. In particular, references

[2,8] develop local AMR or adaptive mesh refinement methods with regard to two dimensional hyperbolic equation sets.

The strong points of AMR methods lie in data structure clarity, the ability to effectively make use of storage, and appropriateness to parallel calculations. The grids which it produces, in universal terms, are changing grids. However, locally, they are uniform grids or meshes on revolving rectilinear subdomains. Users are required to supply difference form programs or procedures on rectilinear subdomain uniform grids or meshes, and these are easy to convert to software.

On the foundation of the AMR programs or procedures in references [2,8], this article referred to Reference [6] to improve
the creation of numerical value flux conservation on the interface surfaces of coarse and fine meshes or grids. This is particularly important with regard to solving discontinuity problems associated with compressible flows. It also makes a number of improvements with regard to such things as boundary treatment, error estimation, and the formation of new grids, etc.

It uses Euler equation sets to calculate Mach 3 air flows in two dimensional tubes having forward facing steps. It carries out discussions with regard to locations of step corners and tangential discontinuities. Numerical value results are very satisfactory.

I. A GENERAL DISCUSSION OF AMR METHODS

This section will outline a discussion of the adaptive grid local refinement methods developed by M.J.Berger. For details, see references [2-5,8].

Adaptive solution processes, for the most part, are composed of time advancement solutions, error estimates, and the formation of new grids. Time advancement solutions begin with a time t. In coarse grids, one first solves for an increment up to $t + \Delta t_c$ (Δt_c is the coarse grid time increment length). Fine meshes adopt a time increment length $\Delta t_f = \Delta t_c / r$ ($r = \Delta x_c / \Delta x_f$ and is the ratio between coarse and fine grids or meshes). Again, one solves, from the time t, for the increment r up $t + \Delta t_c$. The boundary conditions for fine mesh solutions are obtained from physical boundaries (this refers to the boundaries of calculations subdomains), the same intersecting layers of fine meshes or interpolations on coarse grids. Finally, use is made of solutions on fine meshes as weighted averages or interpolations in order to correct solutions on coarse grids for instants $t + \Delta t_c$ associated with overlaid fine meshes.

Due to the fact that subdomains associated with large errors (such as, shock waves, and so on, and so on) will change following along with the advancement of time, as a result of this, between each fixed time increment length, it is necessary to do new estimates of errors in order to form appropriate new fine meshes, and, in conjunction with that, get rid of the old fine meshes. Error estimates are based on Richardson extrapolation automatic advances, and one obtains the main truncation error term τ

$$\tau = \frac{(Q_h^2 - Q_{2h})u(x,t)}{2(2^q - 1)}$$

/332

In this, Q_h is the two layer explicit difference operator. Q_{2h} is the increment length enlarged one fold and associated with the same difference operator. q is the Q_h truncated error order number.

After solving for the truncation error τ, one carries out the formation of new grids or meshes. First of all, one marks the grid or mesh points which exceed given standards with regard to τ. Again, one divides up the marked points into groups so as to obtain a rectilinear fine mesh associated with each drawn up group of marked points which satisfies the given indices or targets in force (the indices or targets in force are the number of marked points in rectangles drawn up as compared to the number of coarse grid points). The objective is to diminish the refined mesh subdomains. Finally, on the four sides of the rectilinear fine meshes obtained from the drawing up, one also obtains an increased buffer zone. Within the time increment for obtaining the next iteration of error estimates, it is necessary for the refinement subdomains to be included right into the fine mesh grids.

Above, we have talked in a general way about coarse and fine two layer grid or mesh adaptive processes. In AMR methods, it is possible to have multiple layers of coarse and fine grids or meshes, that is, on fine meshes, it is also possible to have even finer layers of mesh.

AMR programs or procedures opt for the use of tree mechanisms in their data structures in order to manage the grid or mesh information and relationships. Moreover, use is made of connection tables in order to manage the storage of solutions. All solutions on grids or meshes are distributed and stored in one very large one dimensional number set. This saves on the amount of storage. This being the case, it makes AMR methods, which have complicated logical controls, achieve realization and convenience of maintenance and use.

In AMR methods, see references [3,5] for the stability of one dimensional coarse and fine grid or mesh interface conditions and a theoretical analysis of conservation in them. Reference

[5] only derives interface conservation forms for special cases. With regard to the general case, it only gives a theoretical procedure. In recent years, although work in this area has advanced, as far as coarse and fine grid or mesh interfaces associated with an arbitrary direction are concerned, their conservative nature has still been difficult to show, however.

II. IMPROVEMENTS IN AMR METHODS

We know that maintaining conservation in the natures of truncation solution calculations is unusually important. As far as AMR methods are concerned, besides opting for the use of conservation type difference forms, it is also necessary to maintain conservation on coarse or fine grid or mesh interfaces within calculation subdomains. In references [2,8], AMR methods, due to definitions of unknown quantities at grid or mesh points must think about conservation, and, it is necessary to supply an extra conservation form associated with coarse and fine grid or mesh interfaces, and, again, because of the arbitrary direction of the grids or meshes, it is very difficult to realize conservation as a result [5]. Reference [7], for steady Euler equation sets, realized numerical value flux conservation on coarse of fine grid or mesh interfaces with consistent directions. After it, Reference [6] realized conservation for unsteady Euler equation sets. As far as our reference to [6] is concerned, we took AMR calculation methods and advanced them to have numerical value flux conservation on coarse and fine grid or mesh interfaces.

For the sake of easily realizing conservation, we set the limits below on grids or meshes. Each grid or mesh may not have an arbitrary direction. Fine mesh boundaries and coarse grid lines coincide. Except for physical boundaries, coarse and fine grid or mesh layer boundaries are not allowed to coincide. Here,

grid or mesh layer boundaries stand for all boundaries, except for overlap sections. Unknown quantities are defined at the center of grid lattices.

Below, we use a two layer explicit conservation type difference form in order to conduct discussions. It can be written in a unified form

$$u_{ij}^{n+1}=u_{ij}^{n}-\frac{\Delta t}{\Delta x}(F_{i+\frac{1}{2}j}^{n}-F_{i-\frac{1}{2}j}^{n})-$$

$$-\frac{\Delta t}{\Delta y}(G_{ij+\frac{1}{2}}^{n}-G_{ij-\frac{1}{2}}^{n})$$

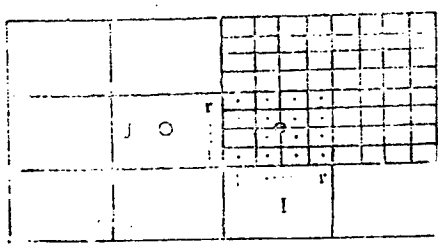


Fig.1 The Interface Between the Coarse and Fine Grids

We first discussed using a solution on a coarse grid (Fig.1) with corrections from a fine mesh overlaid on top of it. Assume that the coarse/fine grid/mesh ratios are

$$\Delta x_c/\Delta x_f=\Delta y_c/\Delta y_f=\Delta t_c/\Delta t_f=r$$

Use the fine mesh solution $\{u_{ij}^{f}\}_{i,j=1}^{r}$ within the coarse grid lattice (I,J) in order to weight averages substituting u_{IJ}^{c}

/333

$$u_{I,J}^{c}\longleftarrow\sum_{i=1}^{r}\sum_{j=1}^{r}w_{ij}\,u_{ij}^{f},\quad\sum_{i=1}^{r}\sum_{j=1}^{r}w_{ij}=1$$

One has

$$u_{I,J}^{c}(t+\Delta t_c)=\sum_{i=1}^{r}\sum_{j=1}^{r} w_{ij}\, u_{ij}^{f}(t+r\,\Delta t_f)$$

$$=u_{I,J}^{c}(t)-\frac{\Delta t_c}{\Delta x_c}\left[\sum_{k=0}^{r-1}\sum_{j=1}^{r} w_{rj}\, F_{r+\frac{1}{2}j}^{f}(t+k\,\Delta t_f)\right.$$

$$\left.-\sum_{k=0}^{r-1}\sum_{j=1}^{r} w_{1j}\, F_{\frac{1}{2}j}^{f}(t+k\,\Delta t_f)\right]$$

$$-\frac{\Delta t_c}{\Delta y_c}\left[\sum_{k=0}^{r-1}\sum_{i=1}^{r} w_{ir}\, G_{ir+\frac{1}{2}}^{f}(t+k\,\Delta t_f)\right.$$

$$\left.-\sum_{k=0}^{r-1}\sum_{i=1}^{r} w_{i1}\, G_{i\frac{1}{2}}^{f}(t+k\,\Delta t_f)\right]$$

$$-\frac{\Delta t_c}{\Delta x_c}\sum_{k=0}^{r-1}\sum_{i=1}^{r-1}\sum_{j=1}^{r}(w_{ij}-w_{i+1j})\,F_{i+\frac{1}{2}j}^{f}(t+k\,\Delta t_f)$$

$$-\frac{\Delta t_c}{\Delta y_c}\sum_{k=0}^{r-1}\sum_{i=1}^{r}\sum_{j=1}^{r-1}(w_{ij}-w_{ij+1})\,G_{ij+\frac{1}{2}}^{f}(t+k\,\Delta t_f)$$

It is possible to see that, in order to maintain coarse and fine grid and mesh numerical value flux conservation, one selects, on lattice sides, corresponding fine mesh numerical value flux sums from t to t = (r-1) Δt_f.

At places where coarse and fine grid or mesh interfaces overlap (I-1/2, J), in order to cause conservation of flux, it is possible to correct the values for places where fine mesh is at $\{(1,j)\}_{j=1}^{r}$ or correct the values for places where coarse grids are at the position (I-1,J). Reference [7] discovered that corrected fine mesh values are unstable. Moreover, corrected coarse grid values are not only stable but highly accurate. From the discussion above, one knows that numerical value flux at locations on boundaries (I-1/2,J) is

$$\tilde{F}_{I-\frac{1}{2}J}=\frac{1}{r^2}\sum_{k=0}^{r-1}\sum_{j=1}^{r} F_{\frac{1}{2}j}^{f}(t+k\,\Delta t_f)$$

Because of this, the corrected form is

$$u^c_{I-1,J}(t+\Delta t_c)=u^c_{I-1,J}(t)-\frac{\Delta t_c}{\Delta x_c}[\tilde{F}_{I-\frac{1}{2},J}-F^c_{I-\frac{3}{2},J}(t)]$$

$$-\frac{\Delta t_c}{\Delta y_c}[G^c_{I-1,J+\frac{1}{2}}(t)-G^c_{I-1,J-\frac{1}{2}}(t)]$$

In programs or procedures, the flux values which are required for correction have already all been automatically stored. Users have no need to consider problems of conservation.

We are limited to the fine mesh boundary types being the same within the same coarse grid lattice. Because this is the case, these fine mesh boundary values are capable of being obtained all at one time. This saves on the amount of work. Also, as far as stored boundary form information is concerned, because this is the case, it is possible to do special handling of forms on physical boundaries. /334

For the sake of flexibility of use, we increased capability for indicated mesh refinement in any subdomain of interest as well as the capability of delimiting subdomains where it was possible to refine the fine mesh. We took the procedure for adding buffer zones in the formation of new grids or meshes and placed it after the completion of error estimates for execution. In this way, the added buffer zones are only added on the four sides all around. The overlap areas on the same level are not increased.

The original AMR program [2,8] is approximately 3 thousand FORTRAN sentences. The program after improvements is approximately 5 thousand sentences (this does not include explanations and solution forms). Users do not need to completely read and understand the AMR program. Moreover, it is only necessary to supply algorithm subroutines and physical boundary subroutines for it to work.

III. CALCULATION CASES AND NUMERICAL VALUE RESULTS

The calculation cases are Mach 3 flow movement problems [25] in two dimensional tubes with forward facing steps. As far as unsteady Euler equation sets are concerned, option is made for the use of classical MacCormack explicit two step difference forms with the addition of artificial viscosity [26]. In these calculation cases, the corners of forward facing steps are the centers of thinned out waves. They are points of singularity. In the vicinity, they give rise to very large numerical value dispersion errors. This leads to shock waves showing Mach reflection at the steps. Reference [25] carried out iso-entropic iso-enthalpic calibration treatments on grid points in the vicinities of step corners. Basically, this caused the steps to be regular shock wave reflectors. We hoped to be able, through refinements of the fine mesh in the vicinity of the step corners, to dissipate dispersion errors in order to arrive at the same kind
of results.

Selecting the time increment length to be 0.8, the factor takes advantage of the maximum permissible time increment length. The first layer basic grid or mesh adopts $\Delta x = \Delta y = \Delta = 1/10$. In order to reduce the amount of calculations, second layer grid or mesh refinements are only indicated for subdomains of interest ($0 \leq x \leq 1.6$) ($\Delta = 1/20$). From $t = 0.0$ to 2.0, each interval of the second layer grid or mesh was 8 increments. For $t = 2.0$ up to 4.0, each interval was 16 increments estimating errors once. Adaptation was used to form the third layer mesh ($\Delta = 1/80$). The error estimation standard was taken to be 0.0075. Buffer zones were taken to be one grid or mesh distance. The effectiveness index of groups of marked points was taken to be 0.75.

Fig.2-4 give isodensity lines for $t = 0.5$, 2.0, and 4.0 instants. In the figures, the first layer grid or mesh has two:

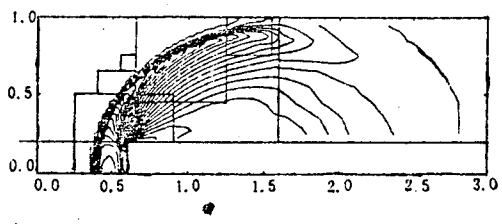


Fig.2 Iso-Density
0.3158(0.1819)5.5934,
N=34,T=0.5

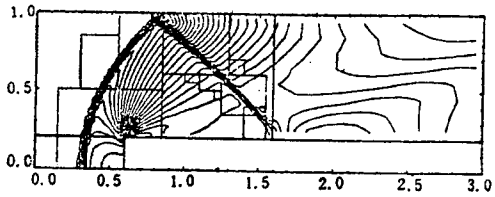


Fig.3 Iso-Density
0.3729(0.2241)6.8737,
N=129,T=2.0

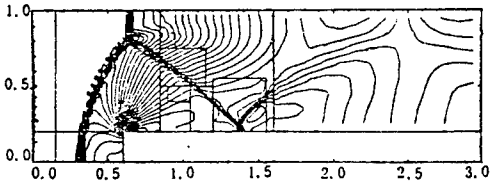


Fig.4 Iso-Density
N=250,T=4.0

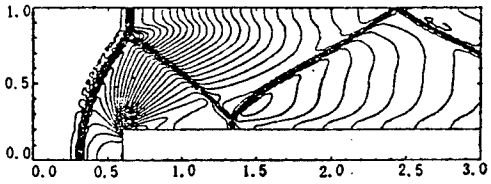


Fig.5 Iso-Density
N=2264,T=4.0

/335

[0,3]x[0.2,1] and [0,0.6]x[0,0.2]. The second layer gird or mesh also has two: [0,1.6]x[0.2,1] and [0,0.6]x[0,0.2]. The remaining rectangles all are third layer grids or meshes. In the figures, N indicates the number of first layer grid or mesh time increments. T indicates time. Taking the results for the finest layer meshes, and comparing them with the results for uniform girds or meshes of the same class (Δ = 1/80) (Fig.5). AMR results and results for uniform grids or meshes agree with each other. However, the CPU time required by AMR is only 28% of that for uniform grids or meshes (Δ = 1/80). The calculation results above have all been corrected at the corners of the steps. If corrections are not done, then, on the steps, there is

clear Mach reflection. Fig.6 is the uniform grid or mesh results (Δ = 1/80).

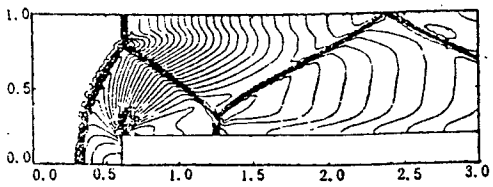


Fig.6 Iso-Density
0.6254(0.1926)6.2112,N=2034,T=4.0

Again, in the vicinity of the step corners, we specify a fourth layer mesh refinement (Δ = 1/320), [0.55,0.65]x[0.2,0.25] and [0.55,0.6]x[0.15,0.2]. The remaining parameters are the same as those selected in Fig.4. Fig.7 is the results for the corresponding calculations as compared to Fig.4-5. It is possible to see that the results using correction treatments are consistent. This explains the definite ability of reducing dispersion errors at step corners to diminish singularity. The amount of operations required is 45% of the uniform grid or mesh (Δ = 1/80).

Due to the fact that tangential discontinuity strengths are weak when compared with shock waves, there is no differentiable outcome and we carried out refinement. At present, for this subdomain, we specified the refinement of a fourth layer mesh (Δ = 1/320). There is a total of 4 grids or meshes: [0.55,0.7875]x[0.725,0.9], [0.725,1.025]x[0.75,0.925], [1.025,1.2625]x[0.775,0.9375], and [1.2625,1.5]x[0.8,0.9375]. From Fig.7's t = 4.0 calculations were done up to 4.5. Fig.8 is the calculation results. Tangential discontinuities are clearly distinguishable.

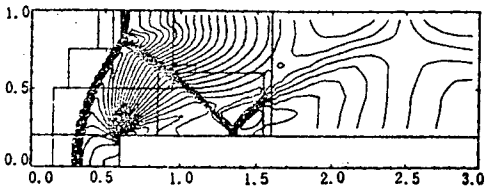


Fig.7 Iso-Density 0.2362
(0.2007)6.0574,N=251,T=4.0

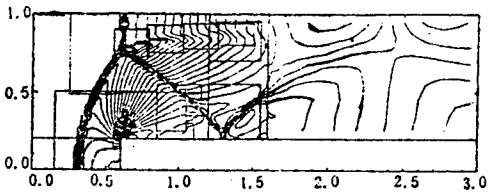


Fig.8 Iso-Density 0.2469
(0.2167)6.5325,N=282,T=4.5

IV. CONCLUSIONS

We improved the AMR programs of M.J.Berger and J.Oliger [2,8] with reference to M.J.Berger and P.Colella [6]. We realized numerical value flux conservation on the interfaces of coarse and fine grids and meshes. The work was appropriate for use to calculate solutions for compressible flow fields with discontinuities. Numerical value calculation cases also prove that it is possible to obtain accurate shock wave results. We also made some improvements in boundary treatment, error estimation, and the formation of new grids and meshes. We added in the specification of subdomain refinement functions and were able to do detailed studies of refinement in any subdomain of interest. During calculations of Mach 3 flow movements in two dimensional tubes having forward facing steps, the amount of operations required for AMR methods is only 28% of the uniform grid of the same class. We specified refinement with a very fine mesh in the vicinity of step corners. As a result of this, there was a reduction in dispersion errors, and we obtained and used shock wave reflections which were consistent with correction treatments. The operations needed was also only 45% of the uniform grid or mesh. Again, for the locations of tangential discontinuities, we specified refinements with very fine grids or meshes and made them clearly distinguishable.

REFERENCES

[1] Bai,D., Brandt, A., SIAM J. Sci. Stat. Comput., 8 (1987), 109-134.

[2] Berger, M.J., Adaptive mesh refinement for hyperbolic partial differential equations, Ph.D. dissertation, Computer Science Dept., Stanford U., U.S.A., (1982).

[3] Berger, M.J., Math. Comput., 45 (1985), 301-318.

[4] Berger, M.J., SIAM J. Sci. Stat. Comput., 7 (1986), 904-916.

[5] Berger, M.J., SIAM J. Numer. Anal., 24 (1987), 967-984.

[6] Berger, M.J., Colella, P., J. Comput. Phys., 82 (1989), 64-84.

[7] Berger, M.J., Jameson, A., AIAA J., 23 (1985), 561-568.

[8] Berger, M.J., Oliger, J., J. Comput. Phys., 53 (1984), 484-512.

[9] Brackbill, J.U., in Numerical Grid Generation, editor J.F. Thompson, North-Holland, New York, (1982), 277-294.

[10] Bolstad, J.H., An adaptive finite difference method for hyperbolic systems in one space dimension, Ph.D. dissertation, Computer Science Dept., Stanford U., U.S.A., (1982).

[11] Caruso, S.C., Adaptive grid techniques for elliptic fluid-flow problems, Ph.D. dissertation, Dept. of Mechanical Engineering, Stanford U., U.S.A., (1985).

[12] Caruso, S.C., Ferziger, J.H., Oliger, J., AIAA Paper 86-0498.

[13] Chan, T.F., Glowinski, R., Periaux, J., Widlund, O.B. (eds.), Domain Decomposition Methods, Proceedings of the Second International Symposium on Domain Decomposition Methods, Los Angeles, California, January 14-16, (1988).

[14] Dwyer, H.A., AIAA J., 22 (1984), 1705-1712.

[15] Dwyer, H.a., Smooke, M.D., Kee, R.J., in Numerical Grid Generation, editor J.F. Thompson, North-Holland, New York, (1982), 339-356.

[16] Glowinski, R., Golub, G.H., Meurant, G.A., Periaux, J. (eds), Domain Decomposition Methods for PDE's, First International Symposium on Domain Methods for Partial Differential Equations, SIAM, Philadelphia, (1988).

[17] Gnoffo, P.A., in Numerical Grid Generation, editor J.F. Thompson, North-Holland, New York, (1982), 819-836.

[18] Gropp, W.D., SIAM J. Sci. Stat. Comput., 1 (1980), 191-197.

[19] Gropp, W.D., SIAM J. Sci. Comput., 8 (1987), 292-304.

[20] Murman, E.M., Baron, J.R., Computational methods for complex flowfields, (1984) Annual Report, AD-A145 219, or AFOSR-TR-83-0841.

[21] Nakahashi, K., Diewert, G.S., AIAA Paper 85-0486.

[22] Saltzman, J., Brackbill, J., in Numerical Grid Generation, editor J.F. Thompson, North-Holland, New York (1982), 865-884.

[23] Skamarock, W., Oliger, J., Street, R.L., J. Comput. Phys., 80 (189), 27-60.

[24] Thompson, M.C., Ferziger, J.H., J. Comput. Phys., 82 (1989), 94-121.

[25] Woodward, P., Colella, P.J. Comput. Phys., 54 (1984), 115-173.

[26] Li Songbo; ACTA AERODYNAMICA SINICA, 1, 4(1983),1-9

DISTRIBUTION LIST

DISTRIBUTION DIRECT TO RECIPIENT

ORGANIZATION	MICROFICHE
B085 DIA/RTS-2FI	1
C509 BALLOC509 BALLISTIC RES LAB	1
C510 R&T LABS/AVEADCOM	1
C513 ARRADCOM	1
C535 AVRADCOM/TSARCOM	1
C539 TRASANA	1
Q592 FSTC	4
Q619 MSIC REDSTONE	1
Q008 NTIC	1
Q043 AFMIC-IS	1
E051 HQ USAF/INET	1
E404 AEDC/DOF	1
E408 AFWL	1
E410 AFDTC/IN	1
E429 SD/IND	1
P005 DOE/ISA/DDI	1
P050 CIA/OCR/ADD/SD	2
1051 AFIT/LDE	1
P090 NSA/CDB	1
2206 FSL	1

Microfiche Nbr: FTD94C000491
NAIC-ID(RS)T-0921-92